Système de détection d'intrusions

SOMMAIRE

1. Contexte et cahier des charges

VIRONAX est acteur français dans la recherche de vaccins. C'est une PME récente d'une cinquantaine de personnes.

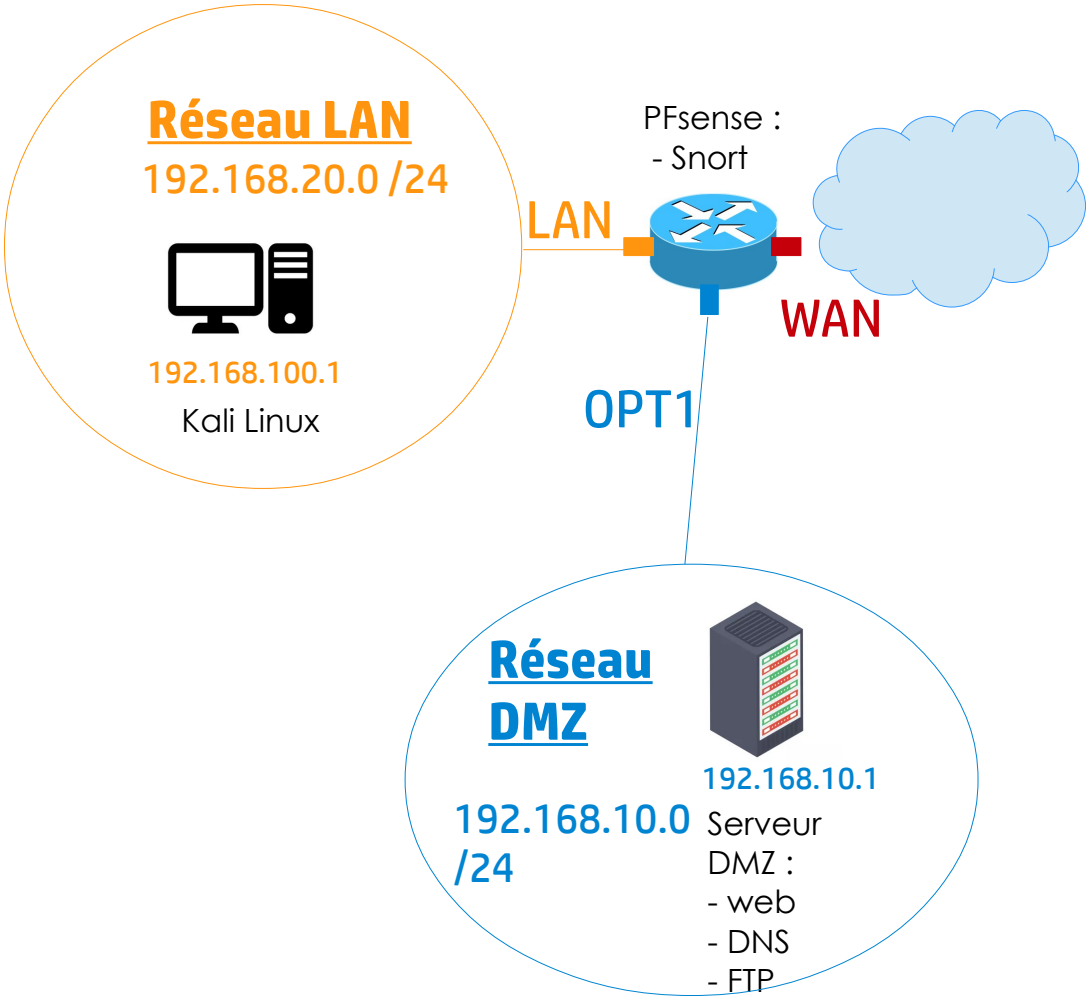
Nous nous trouvons dans la petite équipe du service informatique. Notre maitre de stage, qui n'a pas encore installé d'IDS, nous missionne pour faire des recherches sur des IDS, que nous en choisissons un et que nous fassions des tests de pénétration pour voir comment le logiciel réagit.

Notre tuteur nous demande pour l'instant de créer des scénarios pour 2 types d'attaques informatiques et d'en faire une documentation.

1.1. Objectifs

- Définir ce qu'est un IDS et en sélectionner un
- Définir 2 types d'attaques informatiques
- Proposer et simuler des scenarii d'attaques pour ces piratages
- Voir comment l'IDS réagit pour chaque scénario
- Rédiger une documentation accessible par tous

1.2. Architecture



2. Qu'est-ce qu'un IDS ?

IDS, "**Intrusion detection System**" est un **système de détection d'intrusion**. Il permet de surveiller le trafic réseau, de détecter des activités suspectes voire malveillantes ou encore des violations des politiques de sécurité et d'en faire une alerte. L'administrateur lit les résultats et décide ce qui doit être mis en place face à une ou des attaques données.

Il existe deux catégories d'IDS :

- détection par signatures : reconnaissance de comportements de programmes malveillants par analyse des paquets réseau. Ce système maintient une base de données signatures d'attaque avec laquelle il compare les paquets réseau. Les dernières attaques non répertoriées peuvent échapper à sa vigilance.
- détections par anomalies : compare l'activité du réseau à un modèle de référence. Le système détecte et signale les moindres écarts. Il fait appel à l'apprentissage automatique pour affiner en permanence le modèle de référence. Ce système peut être facilement sujet aux faux positifs (exemple : un utilisateur autorisé qui vient de se connecter pour la première fois à une ressource).

Nous pouvons installer deux types d'IDS. Un système de détection d'intrusion réseau (**NIDS**), qui est installé à un point stratégique du réseau. Il surveille le trafic entrant et sortant vers les appareils du réseau. Le deuxième est un système de détection d'intrusion hôte (**HIDS**). Il est installé sur un poste spécifique (ordinateur, routeur, serveur) et surveille uniquement l'activité de l'appareil sur lequel il est installé.

Un IDS peut donc aider à accélérer et automatiser la détection de menaces sur le réseau. Il peut aussi aider dans le respect de la conformité des réglementations.

3. Choix de l'IDS

SURICATA : est un open source qui utilise la détection par signatures. Suricata a été développé plus récemment que Snort et il a donc plus de fonctionnalités que celui-ci. Il possède un moteur IDS mutlithread supportant le langage de signature de Snort. Il a aussi une faculté permettant d'extraire des fichiers pour les analyser ultérieurement et il détecte automatiquement les protocoles.

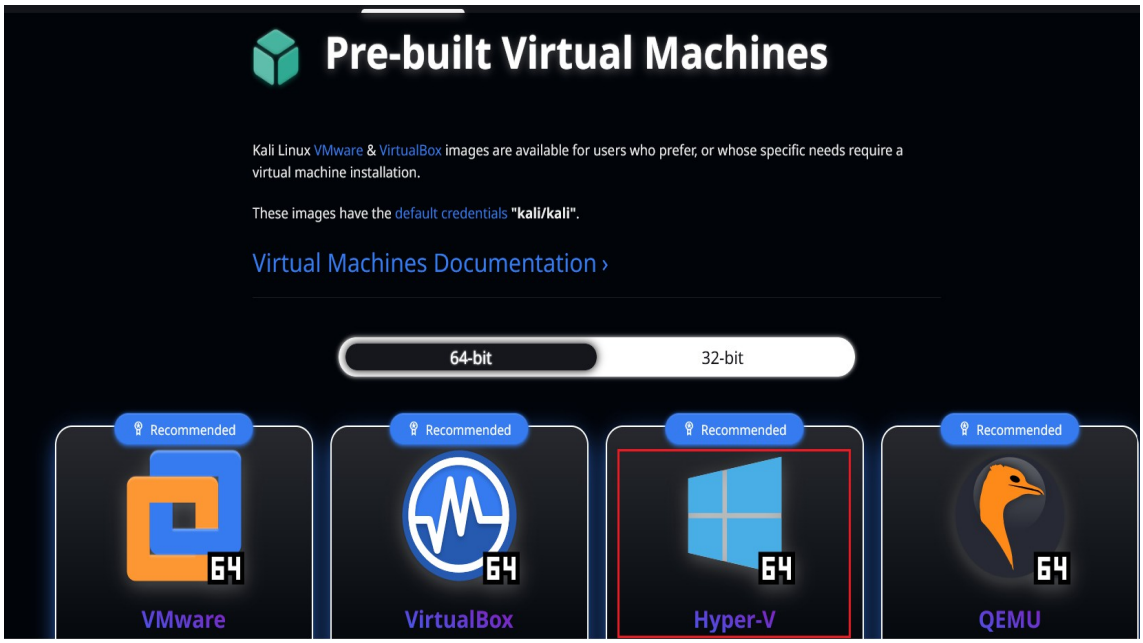
Cet IDS est aussi un IPS (système de prévention et d'intrusion qui analyse également les paquets mais peuvent également les bloquer, ce qui peut stopper automatiquement certaines attaques), mais nous allons seulement nous servir de sa fonction IDS dans le projet. Nous pouvons tout de même souligner que si notre tuteur souhaite mettre en place plus tard un IPS, il pourra juste paramétrer un des IDS en place.

4. Prérequis pour la réalisation des scenarii

4.1.Kali Linux

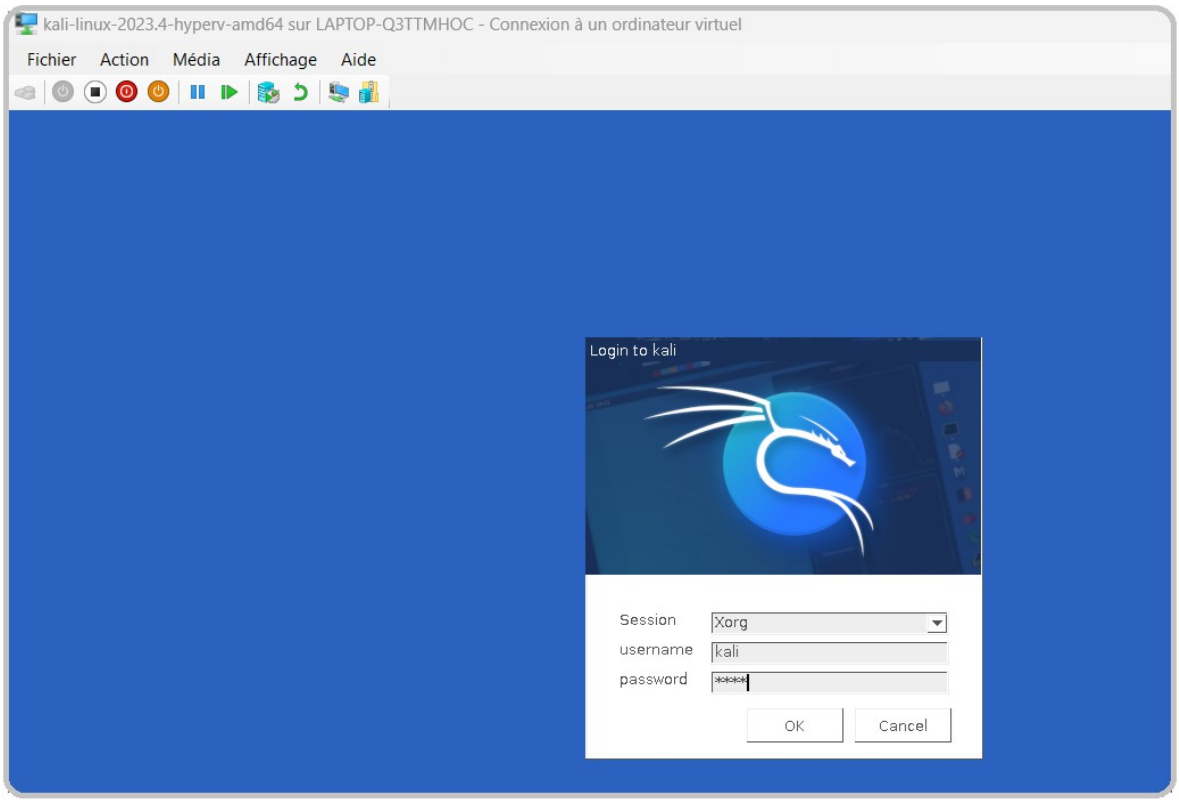
Pour nos tests de pénétration, nous allons utiliser une distribution Linux dans laquelle de nombreux outils facilitant ces tests sont présents : Kali Linux.
Elle a été créée dans le but premier d'identifier des failles de sécurité, de récupérer des données perdues, d'analyser les mots de passe et de décoder des documents chiffrés. Plus de 600 outils de sécurités différents sont proposés avec cette distribution open source. Ce système est considéré très fiable et est même utilisé par des pirates « éthiques ».

Il est possible d'installer KaliLinux à partir d'une image préfaite. Nous utilisons Hyper-V comme virtualiseur. Nous choisissons donc une image KaliLinux préfaite sur Hyper-V.

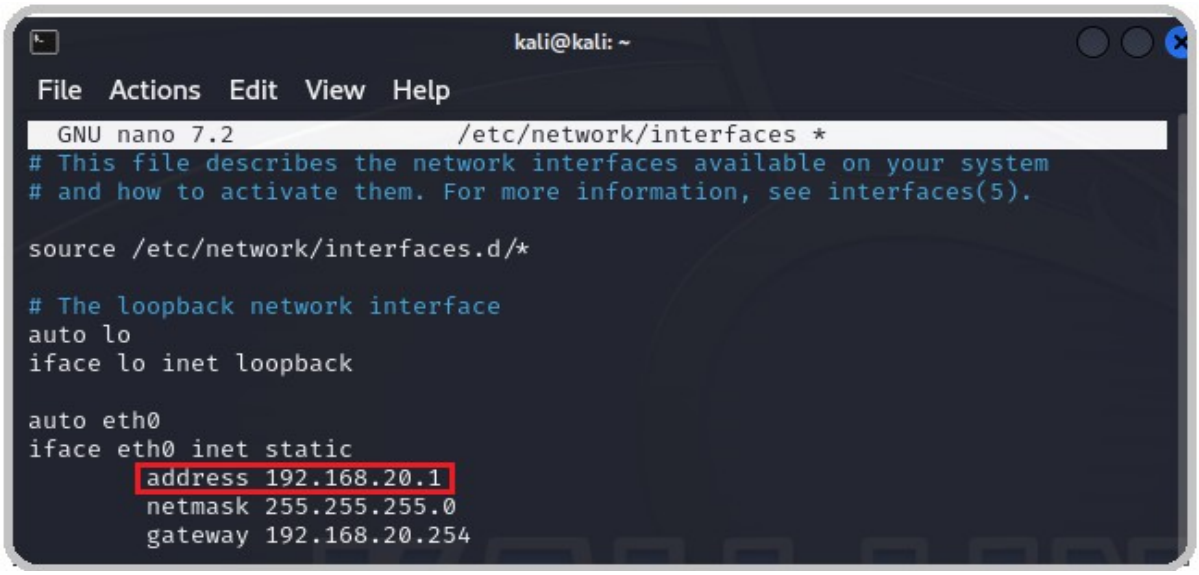


– Après installation, lancement :

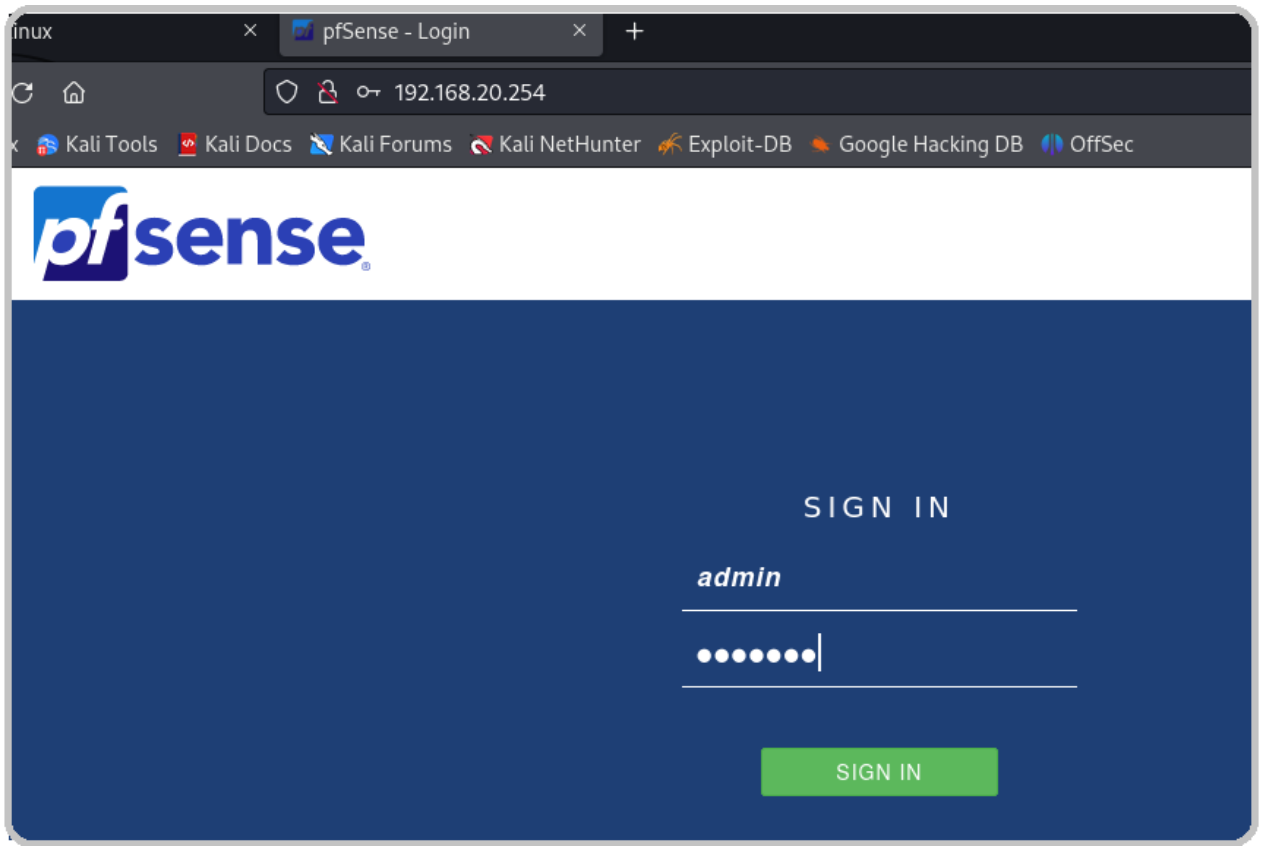
identifiant par défaut : kali
mot de passe par défaut : kali



- Configuration IP



- Connexion à PFsense

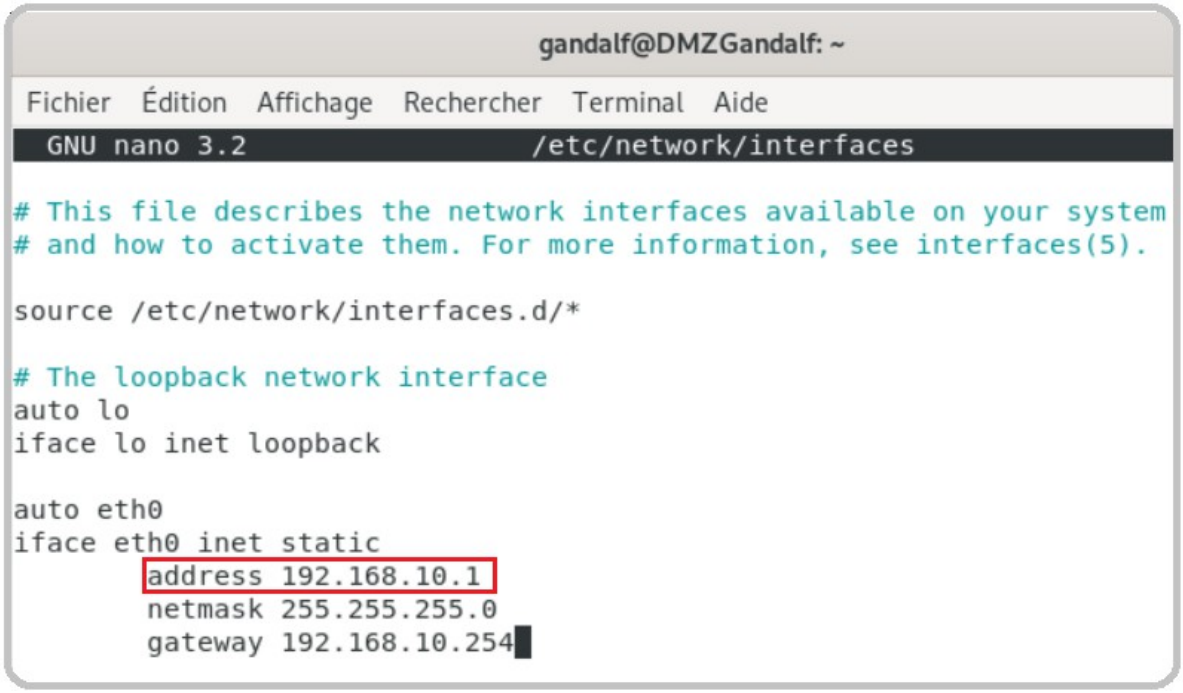


4.2. Machine DMZ

Nous allons à présent installer et configurer notre serveur présent dans le réseau DMZ. Il va regrouper un serveur DNS, un serveur web et un serveur FTP. Nous avons réuni ces trois serveurs sur une seule machine pour plus de facilité mais ils auraient très bien pu être présent sur 3 ordinateurs différents.

Comme l'installation et le paramétrage de ce serveur n'est pas le cœur de l'objectif de ce projet, nous ne vous détaillons pas la procédure.
Nous vous mettons tout de même à disposition les fichiers de configuration et tests de validation pour vous montrer le bon fonctionnement des différents services.

- Configuration IP



- Serveur web avec Apache2

- Serveur DNS

Vérification du bon fonctionnement

- zone directe

```
root@DMZGandalf:/etc/bind# dig DMZGandalf.lotr.lan

; <<>> DiG 9.11.5-P4-5.1+deb10u9-Debian <<>> DMZGandalf.lotr.lan
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 12674
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: e96c7d09c416acccfeef2cf86597c3fd4bf06623b001f32a (good)
;; QUESTION SECTION:
;DMZGandalf.lotr.lan.           IN      A

;; ANSWER SECTION:
DMZGandalf.lotr.lan.    10800   IN      A       192.168.10.1

;; AUTHORITY SECTION:
lotr.lan.               10800   IN      NS      DMZGandalf.lotr.lan.

;; Query time: 0 msec
;; SERVER: 192.168.10.1#53(192.168.10.1)
;; WHEN: ven. janv. 05 09:55:25 CET 2024
;; MSG SIZE  rcvd: 106
```

- zone indirecte

```
root@DMZGandalf:/etc/bind# dig -x 192.168.10.1

; <<>> DiG 9.11.5-P4-5.1+deb10u9-Debian <<>> -x 192.168.10.1
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 61134
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 18321046b0bb92a1c48d144d6597c4cb5db1575bc8596042 (good)
;; QUESTION SECTION:
;1.10.168.192.in-addr.arpa.     IN      PTR

;; ANSWER SECTION:
1.10.168.192.in-addr.arpa. 10800 IN     PTR     DMZGandalf.lotr.lan.

;; AUTHORITY SECTION:
10.168.192.in-addr.arpa. 10800  IN      NS      DMZGandalf.lotr.lan.

;; ADDITIONAL SECTION:
DMZGandalf.lotr.lan.    10800   IN      A       192.168.10.1

;; Query time: 0 msec
```

- ping site internet

```
root@DMZGandalf:/etc/bind# ping www.lotr.lan
PING www.lotr.lan (192.168.10.1) 56(84) bytes of data.
64 bytes from DMZGandalf.lotr.lan (192.168.10.1): icmp_seq=1 ttl=64 time=0.011 ms
64 bytes from DMZGandalf.lotr.lan (192.168.10.1): icmp_seq=2 ttl=64 time=0.050 ms
64 bytes from DMZGandalf.lotr.lan (192.168.10.1): icmp_seq=3 ttl=64 time=0.046 ms
64 bytes from DMZGandalf.lotr.lan (192.168.10.1): icmp_seq=4 ttl=64 time=0.033 ms
64 bytes from DMZGandalf.lotr.lan (192.168.10.1): icmp_seq=5 ttl=64 time=0.054 ms
^C
--- www.lotr.lan ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 91ms
rtt min/avg/max/mdev = 0.011/0.038/0.054/0.017 ms
root@DMZGandalf:/etc/bind#
```

– Serveur FTP

gandalf@DMZGandalf: ~

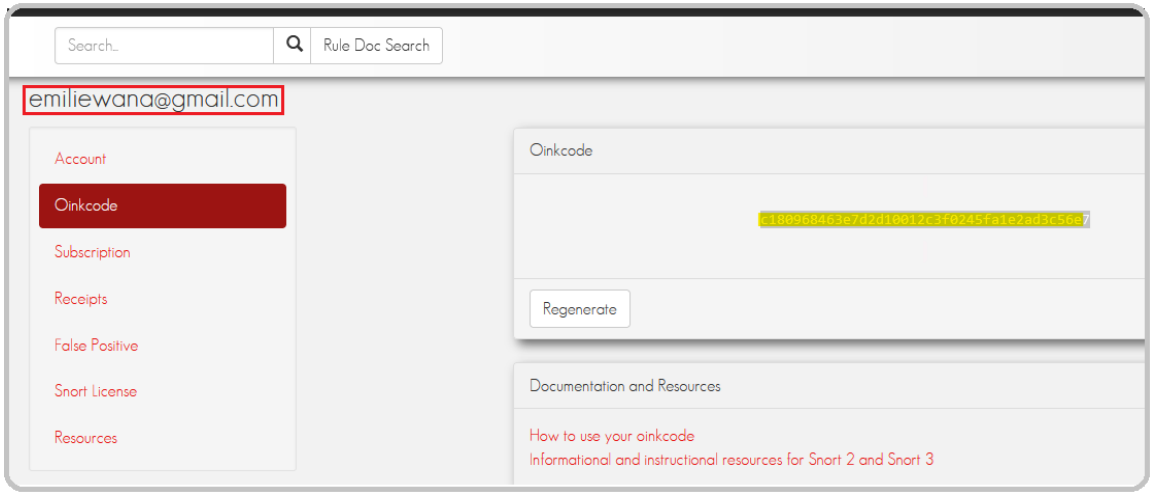
```
  GNU nano 5.4                                ftp-perso.conf *
#Nom du serveur
ServerName "DMZGandalf.lotr.lan"

#Message de connexion
DisplayLogin "La connexion FTP s'est bien effectuée"

#Désactiver IPV6
UseIPV6 off

#Spécifie le répertoire FTP auquel l'utolisateur est autorisé à accéder
DefaultRoot état

#Autoriser la connexion uniquement aux membres du groupe "ftp"
<Limit LOGIN>
Denygroup !ftp
</Limit>
```

4.3. Installation Suricata avec PFsense

Nous téléchargeons et installons Suricata, dans les "available packages" proposés par PFsense.

Paramétrage

- Création d'un compte sur Snort et récupération du « Oinkmaster Code » qui est demandé dans la configuration de Suricata lorsque nous souhaitons activer les règles Snort.



- Dans "Services" / "Suticata" / "Global Settings", nous paramétrons Suricata. Il va s'appuyer sur des règles émises par la communauté Snort et de GeoLite2 pour être plus complet au niveau de la détection de certaines menaces

ESICAD

Interfaces | Global Settings | Updates | Alerts | Blocks | Files | Pass Lists | Suppress | Logs View | Logs Mgmt | SID Mgmt | Sync | IP Lists

Please Choose The Type Of Rules You Wish To Download

Install ETOpen Emerging Threats rules

☑ ETOpen is a free open source set of Suricata rules whose coverage is more limited than ETPro.

☐ Use a custom URL for ETOpen downloads

Enabling the custom URL option will force the use of a custom user-supplied URL when downloading ETOpen rules.

Install ETPro Emerging Threats rules

☐ ETPro for Suricata offers daily updates and extensive coverage of current malware threats.

☐ Use a custom URL for ETPro rule downloads

The ETPro rules contain all of the ETOpen rules, so the ETOpen rules are not required and are disabled when the ETPro rules are selected. Sign Up for an ETPro Account. Enabling the custom URL option will force the use of a custom user-supplied URL when downloading ETPro rules.

Install Snort rules

☑ Snort free Registered User or paid Subscriber rules

Sign Up for a free Registered User Rules Account
Sign Up for paid Snort Subscriber Rule Set (by Talos)

☐ Use a custom URL for Snort rule downloads

Enabling the custom URL option will force the use of a custom user-supplied URL when downloading Snort Subscriber rules.

Snort Rules Filename

snortrules-snapshot-29200.tar.gz

Enter the rules tarball filename (filename only, do not include the URL.)
Example: snortrules-snapshot-29200.tar.gz
DO NOT specify a Snort3 rules file! Snort3 rules are incompatible with Suricata and will break your installation!

Snort Oinkmaster Code

c180968463e7d2d10012c3f0245fa1e2ad3c56e7

Obtain a snort.org Oinkmaster code and paste it here.

Install Snort GPLv2 Community rules

☑ The Snort Community Ruleset is a GPLv2 Talos-certified ruleset that is distributed free of charge without any Snort Subscriber License restrictions.

☐ Use a custom URL for Snort GPLv2 rule downloads

This ruleset is updated daily and is a subset of the subscriber ruleset. If you are a Snort Subscriber Rules customer (paid subscriber), the community ruleset is already built into your download of the Snort Subscriber rules, and there is no benefit in adding this rule set separately.

Install Feodo Tracker Botnet C2 IP rules

☐ The Feodo Botnet C2 IP Ruleset contains Dridex and Emotet/Heodo botnet command and control servers (C&Cs) tracked by Feodo Tracker.

Install ABUSE.ch SSL Blacklist rules

☐ The ABUSE.ch SSL Blacklist Ruleset contains the SSL cert fingerprints of all SSL certs blacklisted by ABUSE.ch.

Hide Deprecated Rules Categories

☐ Hide deprecated rules categories in the GUI and remove them from the configuration. Default is Not Checked.

Download Extra Rules

☐ Download Extra Rules

Download extra rules file or tar.gz archive with rules. If "Check MD5" is set, the code will assume a matching filename exists at the same URL with an additional extension of ".md5".

If you have a subscription for more current GeoIP2 updates, uncheck this option and instead create your own process to place the required database file in /usr/local/share/suricata/GeoLite2/.

GeoLite2 DB Account ID

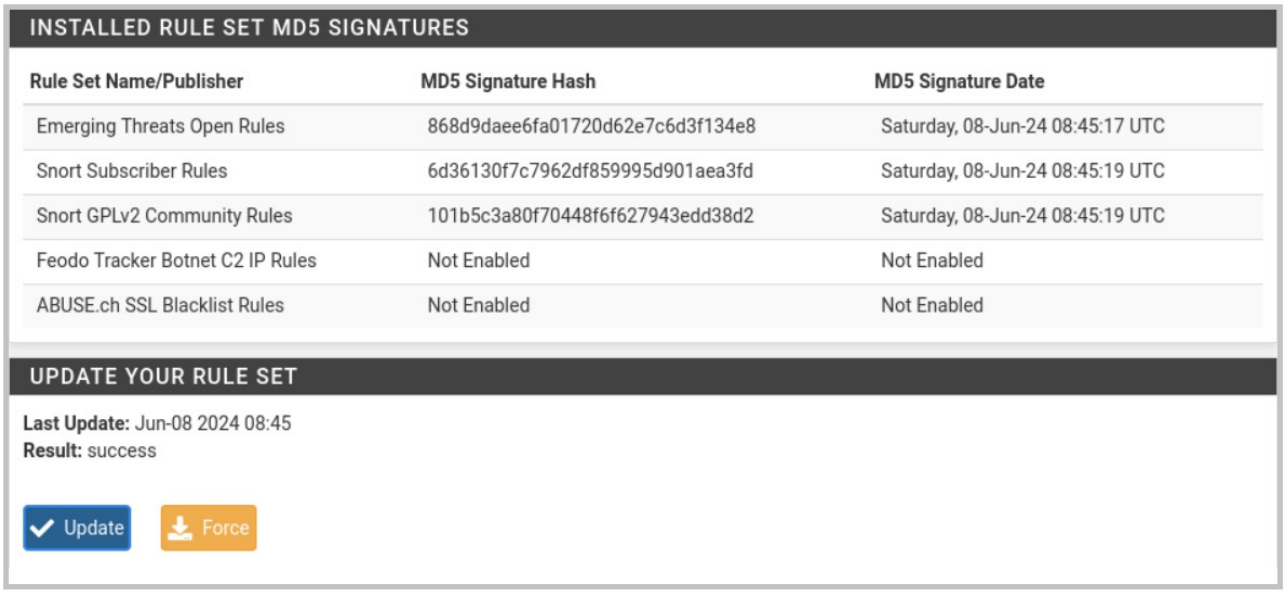
1003950

To utilize the free MaxMind GeoLite2 GeoIP functionality, you must register for a free MaxMind user account. **Use the GeoIP Update version 3.1.1 or newer registration option.**

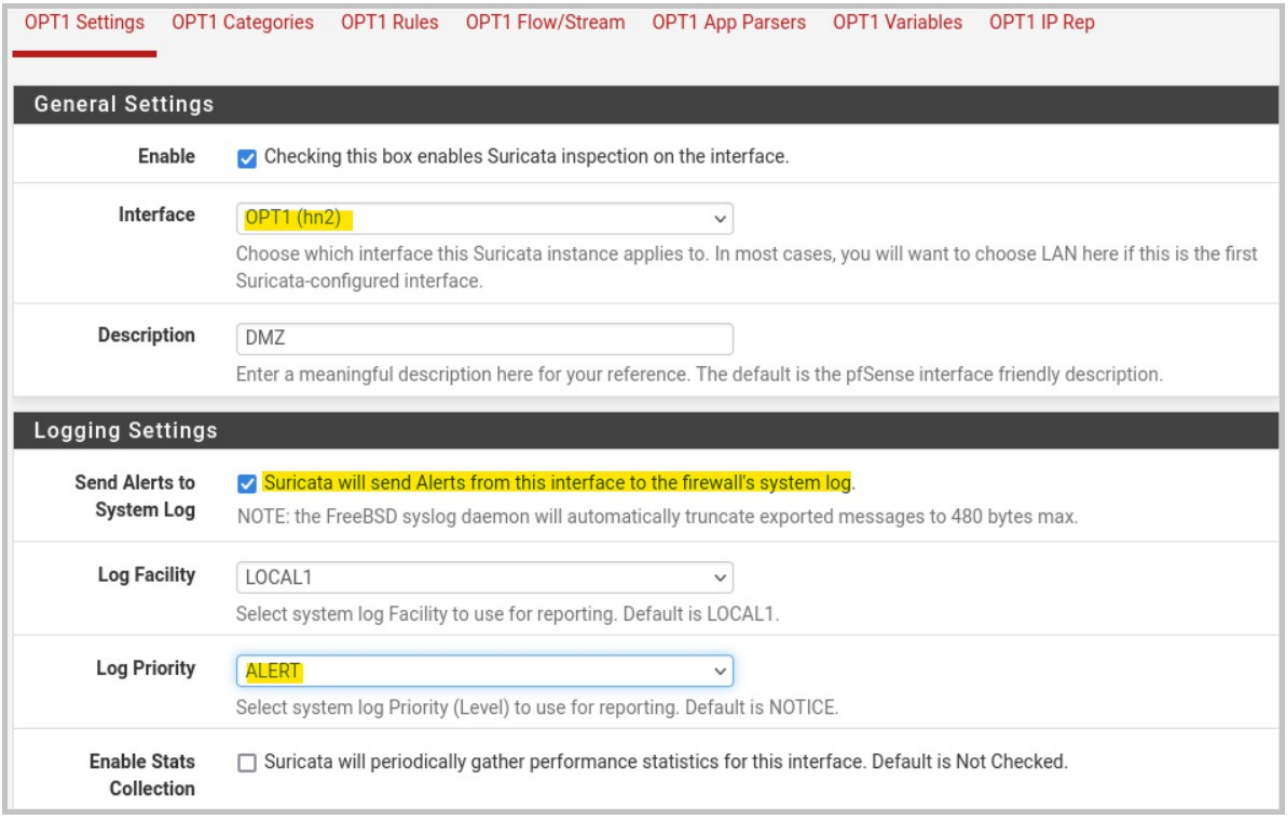
GeoLite2 DB License Key

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

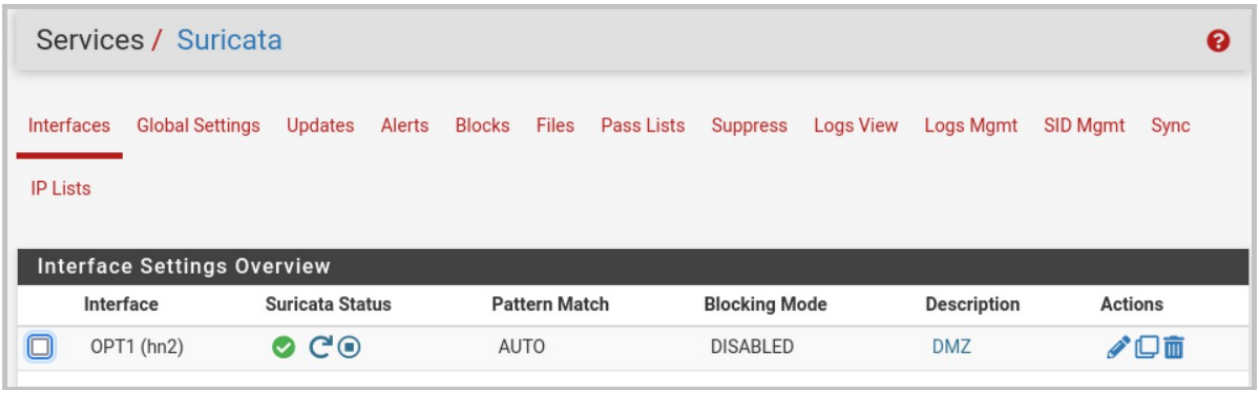
- Dans l'onglet "Update", nous cliquons sur "Update".



- Ajout de l'interface à surveiller, ici OPT1(192.168.10.0/24), là où se trouve notre réseau DMZ qui va subir des attaques. Dans une situation réelle et non de test, nous aurions activé Suricata sur les 3 interfaces car les attaques peuvent venir des interfaces LAN et WAN. Dans ce cas, l'IDS détecte immédiatement et si l'IPS avait été activé, il l'aurait bloquée avant que l'attaque n'arrive sur l'interface OPT1.

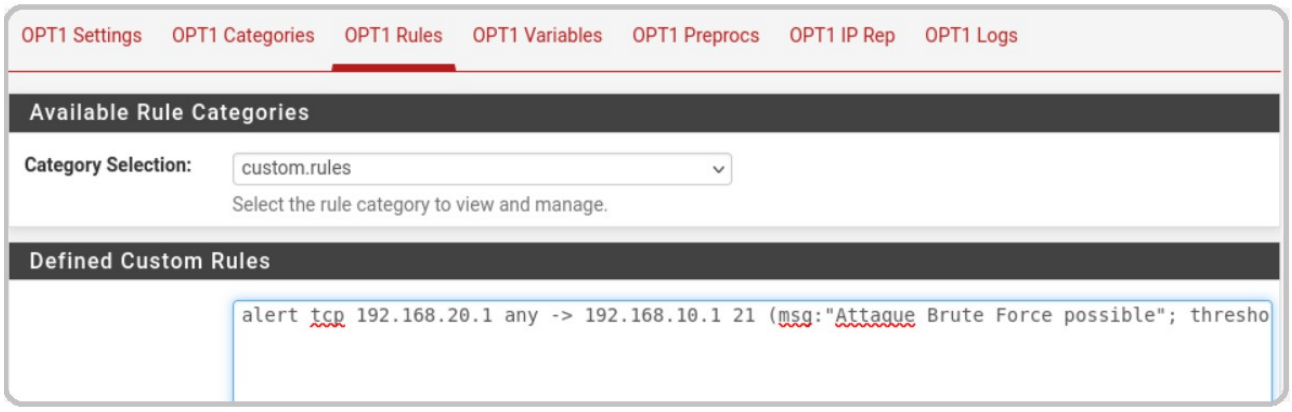


- Démarrage de l'interface



- Pour l'attaque Brute Force, nous allons entrer une règle personnalisée. Pour cela, nous allons dans l'interface Suricata OPT1 / OPT1 Rules et nous sélectionnons la catégorie "custom.rules"
  Nous entrons la règle suivante :

alert tcp 192.168.20.1 any → 192.168.10.1 21 (msg : « Attaque Brute Force possible » ; threshold : type threshold, track by_dst, count 3, seconds 60 ; sid : 1000001;)



- Nous faisons de même pour l'attaque DOS : alert tcp any any -> any any (msg:"DOS possible"; flags: S,12; threshold: type both, track by_dst, count 10, seconds 60; sid: 100035;)

4.4. Utilisation Wireshark

C'est une application open source. C'est un analyseur de réseau qui capte et affiche, par une interface, les données circulant dans le réseau (dans les deux sens).

Un administrateur ou technicien peuvent s'en servir pour déterminer si une machine est infectée ou débugger des problèmes de connectivité grâce à des filtres qui permettent d'affiner la recherche.

Dans notre cas, nous nous servons de Wireshark pour voir ce qui se passe sur le réseau lorsqu'une attaque informatique est lancée.

5. Différents types d'attaques et scenarii

Il existe un très grand nombre de cyberattaques différentes, de la Bruteforce en passant par les malwares ou chevaux de Troie, des injections SQL, etc.
Les objectifs de ces attaques sont différentes comme le vol de données, empêcher l'accès à un service, mais elles mettent toujours à mal le système visé.
Nous allons sélectionner quelques types d'attaques, faire des tests d'intrusion grâce à Kali Linux, ensuite regarder la trame et enfin voir comment notre IDS les détecte.

5.1. Bruteforce

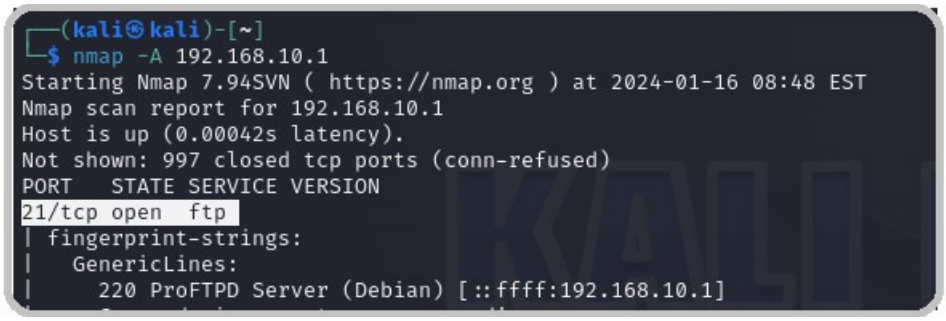
Selon la cnil, dans une "attaque par force brute" (bruteforce attack), le pirate tente de se connecter à un service qui possède un mot de passe ou une clé, en essayant différentes combinaisons possibles jusqu'à ce que cela fonctionne.

Exemple d'attaque avec récupération des identifiants et mot de passe pour le service FTP

Pour effectuer une attaque Bruteforce, nous allons utiliser l'outil **Hydra** déjà installé sous KaliLinux.

Hydra est un outil open source qui permet de forcer des noms d'utilisateurs, de mots de passe de différents protocoles (FTP, SSH, Telnet, SMB...) avec l'aide de dictionnaires de mots ou par essai de tous les caractères possibles.

- Avant d'initier l'attaque, nous utilisons la commande nmap pour vérifier si le port 21 est ouvert.



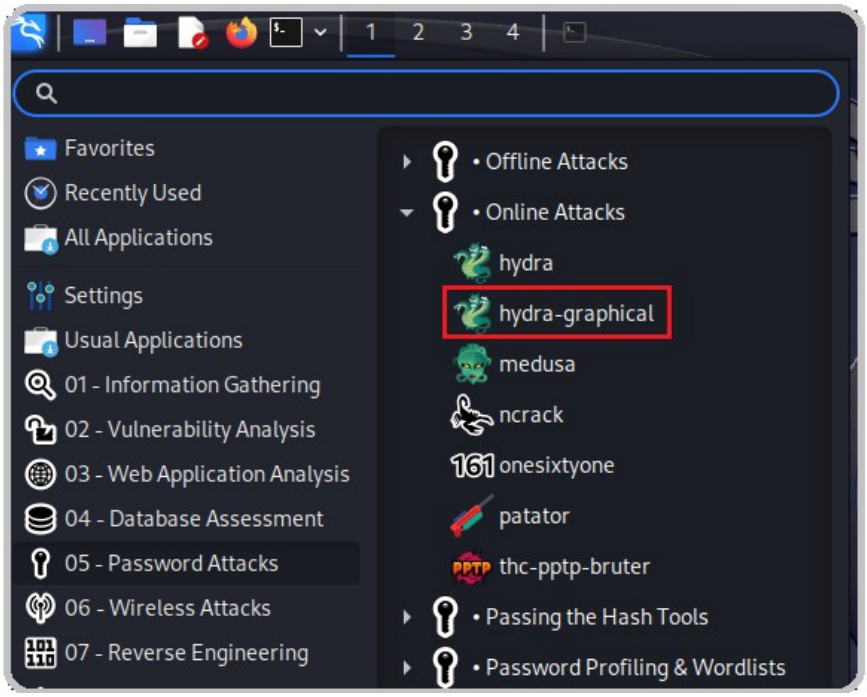
→ c'est le cas

- Sur Suricata, dans les alertes, nous pouvons déjà remarquer des lignes liées à la commande "nmap". Cela est un premier indicateur de l'attaque.

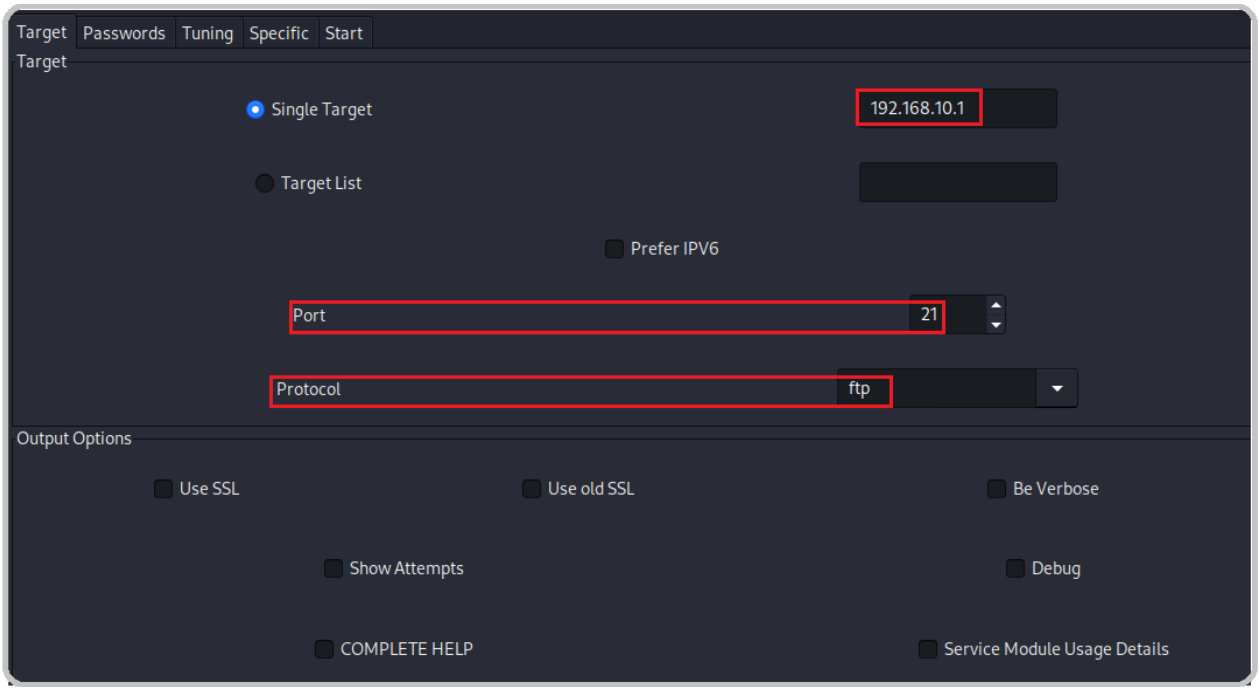
9 Entries in Active Log

Date	Action	Pri	Proto	Class	Source IP	SPort	Destination IP	DPort	GID:SID	Description
2024-01-28 10:23:52	⚠	3	TCP	Unknown Traffic	192.168.20.1	36530	192.168.10.1	80	119:31	(http_inspect) UNKNOWN METHOD
2024-01-28 10:23:16	⚠	2	TCP	Potentially Bad Traffic	192.168.20.1	60982	192.168.10.1	21	125:2	(ftp_telnet) Invalid FTP Command
2024-01-28 10:22:49	⚠	2	TCP	Potentially Bad Traffic	192.168.20.1	41912	192.168.10.1	21	125:2	(ftp_telnet) Invalid FTP Command
2024-01-28 10:22:54	⚠	2	TCP	Potentially Bad Traffic	192.168.20.1	50774	192.168.10.1	21	125:2	(ftp_telnet) Invalid FTP Command
2024-01-28 10:22:44	⚠	2	TCP	Potentially Bad Traffic	192.168.20.1	41910	192.168.10.1	21	125:2	(ftp_telnet) Invalid FTP Command
2024-01-28 10:22:34	⚠	2	TCP	Potentially Bad Traffic	192.168.20.1	49402	192.168.10.1	21	125:2	(ftp_telnet) Invalid FTP Command
2024-01-28 10:21:44	⚠	2	TCP	Potentially Bad Traffic	192.168.20.1	56872	192.168.10.1	21	125:2	(ftp_telnet) Invalid FTP Command
2024-01-28 10:21:39	⚠	2	TCP	Potentially Bad Traffic	192.168.20.1	58194	192.168.10.1	21	125:2	(ftp_telnet) Invalid FTP Command
2024-01-28	⚠	2	TCP	Potentially Bad Traffic	192.168.20.1	58182	192.168.10.1	21	125:2	(ftp_telnet) Invalid FTP Command

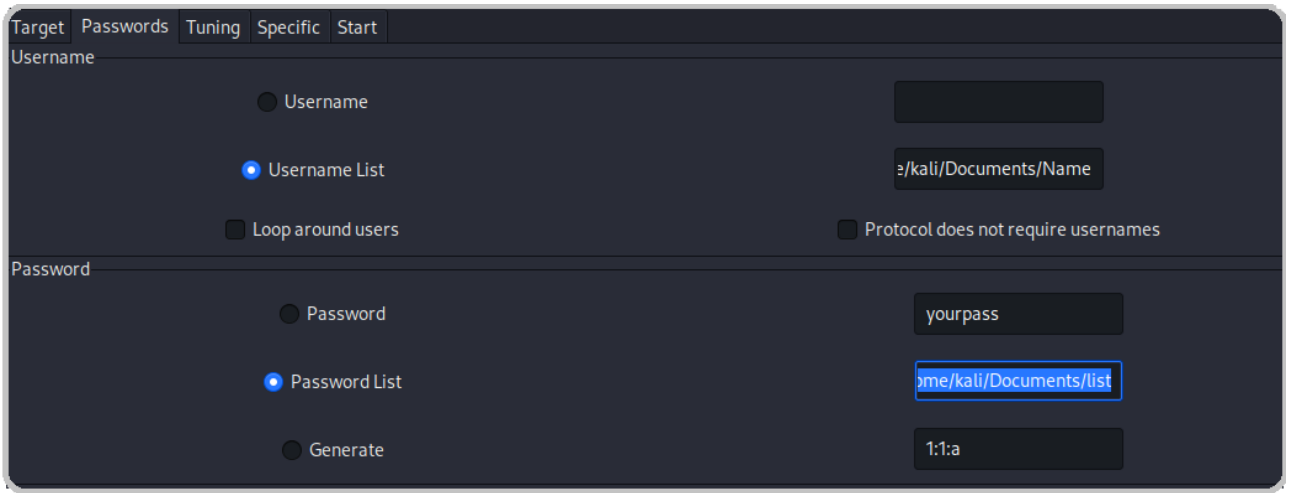
– Ensuite, nous exécutons la version Hydra avec interface

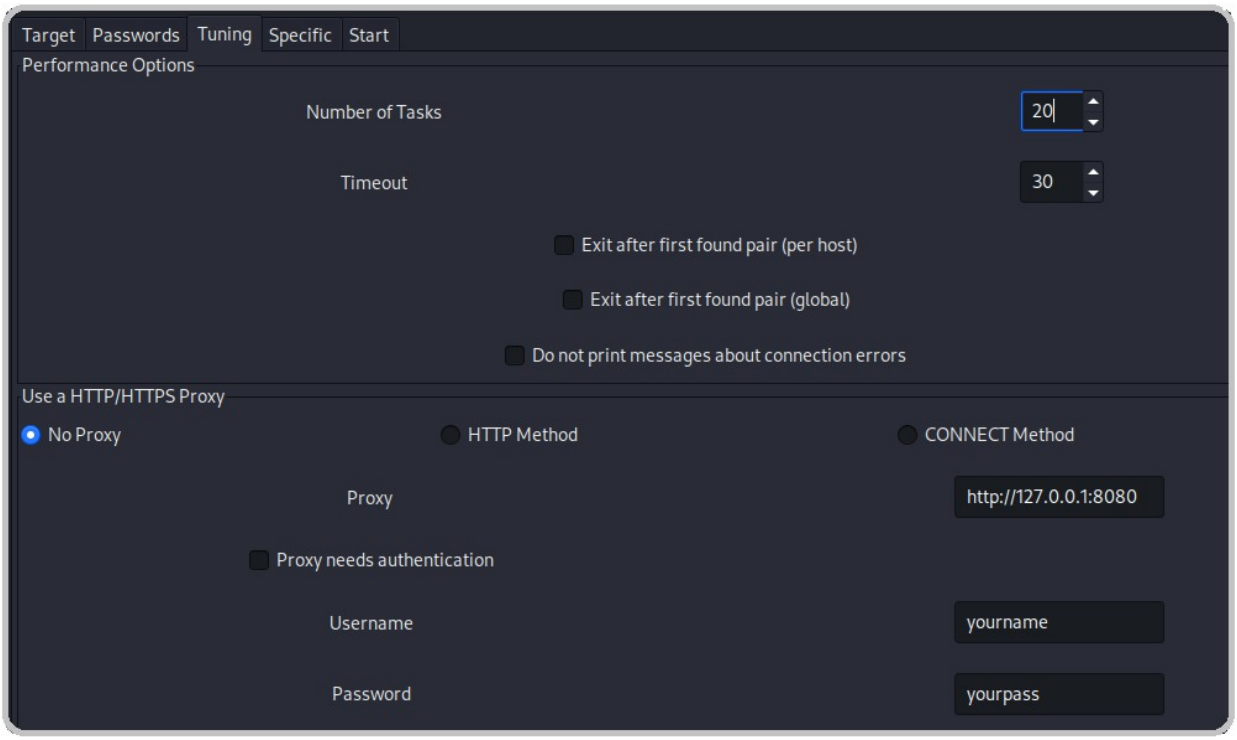


– Nous entrons l'adresse IP de l'ordinateur cible, le port et le protocole

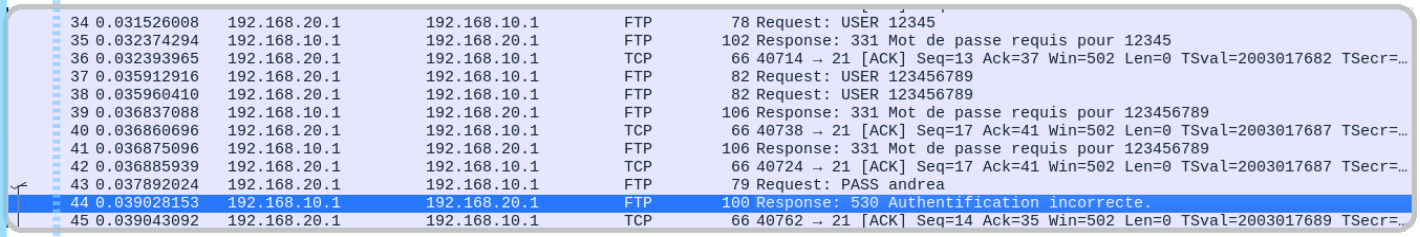


– Ensuite, pour trouver le nom d'utilisateur et le mot de passe, nous nous servons de dictionnaires de mots. Pour notre exemple, nous avons ajouté dans les listes les utilisateurs et les mots de passe pour que le logiciel Hydra les trouve.





- Nous lançons la recherche des identifiants et regardons en parallèle sur Wireshark ce qui se passe.



34	0.031526008	192.168.20.1	192.168.10.1	FTP	78	Request: USER 12345
35	0.032374294	192.168.10.1	192.168.20.1	FTP	102	Response: 331 Mot de passe requis pour 12345
36	0.032393965	192.168.20.1	192.168.10.1	TCP	66	40714 → 21 [ACK] Seq=13 Ack=37 Win=502 Len=0 TSval=2003017682 TSecr=…
37	0.035912916	192.168.20.1	192.168.10.1	FTP	82	Request: USER 123456789
38	0.035960410	192.168.20.1	192.168.10.1	FTP	82	Request: USER 123456789
39	0.036837088	192.168.10.1	192.168.20.1	FTP	106	Response: 331 Mot de passe requis pour 123456789
40	0.036860696	192.168.20.1	192.168.10.1	TCP	66	40738 → 21 [ACK] Seq=17 Ack=41 Win=502 Len=0 TSval=2003017687 TSecr=…
41	0.036875096	192.168.10.1	192.168.20.1	FTP	106	Response: 331 Mot de passe requis pour 123456789
42	0.036885939	192.168.20.1	192.168.10.1	TCP	66	40724 → 21 [ACK] Seq=17 Ack=41 Win=502 Len=0 TSval=2003017687 TSecr=…
43	0.037892024	192.168.20.1	192.168.10.1	FTP	79	Request: PASS andrea
44	0.039028153	192.168.10.1	192.168.20.1	FTP	100	Response: 530 Authentification incorrecte.
45	0.039043092	192.168.20.1	192.168.10.1	TCP	66	40762 → 21 [ACK] Seq=14 Ack=35 Win=502 Len=0 TSval=2003017689 TSecr=…

→ nous remarquons bien qu'Hydra tente différents utilisateurs et mot de passe qui sont pour l'instant incorrects.

- Voici enfin le résultat de la recherche de comptes ftp par Hydra.

```
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-01-16 09:54:03
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session found, to prevent overwriting, ./hydra.restore
[DATA] max 20 tasks per 1 server, overall 20 tasks, 104 login tries (l:8/p:13), ~6 tries per task
[DATA] attacking ftp://192.168.10.1:21/
[21][ftp] host: 192.168.10.1   login: sauron   password: Anneau
[21][ftp] host: 192.168.10.1   login: frodon   password: L'anneauunique
[STATUS] 96.00 tries/min, 96 tries in 00:01h, 27 to do in 00:01h, 1 active
[21][ftp] host: 192.168.10.1   login: gandalf   password: motdepasse
1 of 1 target successfully completed, 3 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-01-16 09:55:44
<finished>
```

- Vérification : ces identifiants sont-ils corrects ?

```
┌──(kali㉿kali)-[/]
└─$ ftp 192.168.10.1
Connected to 192.168.10.1.
220 ProFTPD Server (Debian) [::ffff:192.168.10.1]
Name (192.168.10.1:kali): gandalf
331 Mot de passe requis pour gandalf
Password:
230 Utilisateur gandalf authentifié
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

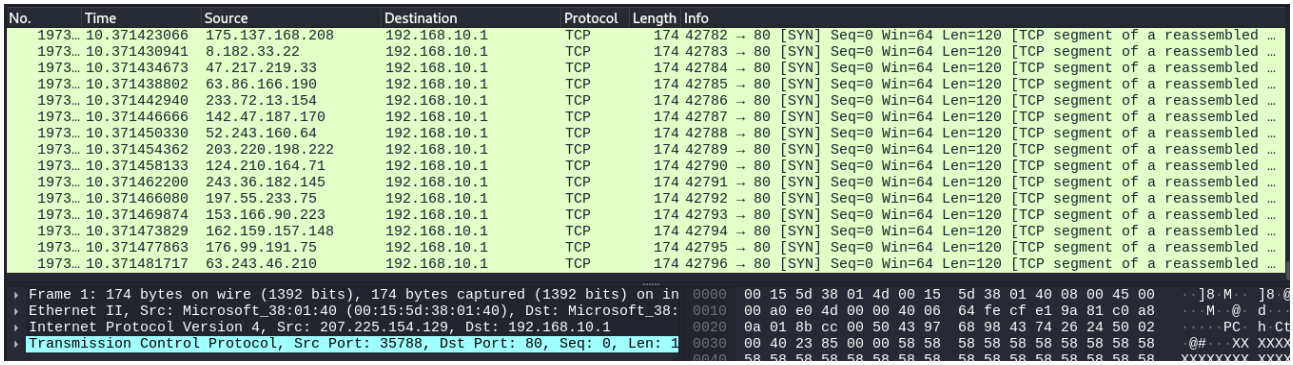
→ oui

- Comment Suricata réagit-il ?

06/08/2024 08:53:38	⚠	3	TCP	Generic Protocol Command Decode	192.168.10.1	21	192.168.20.1	36882	1:2260002	SURICATA Applayer Detect protocol only one direction
06/08/2024 08:53:37	⚠	3	TCP	Not Assigned	192.168.20.1	36916	192.168.10.1	21	1:1000001	Attaque Brute Force possible
06/08/2024 08:53:37	⚠	3	TCP	Not Assigned	192.168.20.1	36884	192.168.10.1	21	1:1000001	Attaque Brute Force possible
06/08/2024 08:53:37	⚠	3	TCP	Not Assigned	192.168.20.1	36846	192.168.10.1	21	1:1000001	Attaque Brute Force possible
06/08/2024 08:53:37	⚠	3	TCP	Not Assigned	192.168.20.1	36878	192.168.10.1	21	1:1000001	Attaque Brute Force possible
06/08/2024 08:53:37	⚠	3	TCP	Not Assigned	192.168.20.1	36802	192.168.10.1	21	1:1000001	Attaque Brute Force possible
06/08/2024 08:53:37	⚠	3	TCP	Not Assigned	192.168.20.1	36870	192.168.10.1	21	1:1000001	Attaque Brute Force possible

Nous venons donc d'effectuer une attaque brute force sur notre serveur DMZ. Cette attaque a réussi car nous avons, grâce à Hydra, trouvé 3 identifiants et mot de passe corrects pour se connecteur au serveur FTP.

5.2. DdoS ou Dos



Une attaque DDoS (Distributed Denial of Service) plus précisément par déni de service distribué consiste en rendre inaccessible un service en envoyant un nombre important de requêtes simultanément pour provoquer une saturation ou en exploitant une faille de sécurité provoquant une panne ou un fonctionnement très dégradé du service.

Les attaques DDos sont similaires aux attaques Dos mais ont quelques différences. Une attaque DDos implique plusieurs systèmes, souvent à l'aide d'un botnet, attaquant un système alors que pour une Dos, un seul système attaque. Une attaque DDos peut être plus rapide car elle envoie plus de requêtes et il est moins facile de détecter d'où elle provient.

Flooding

C'est une forme d'attaque DDoS : saturer un système avec des demandes
Il existe différentes attaques de flood comme le "ping flood", "HTTP flood", "UDP flood"...

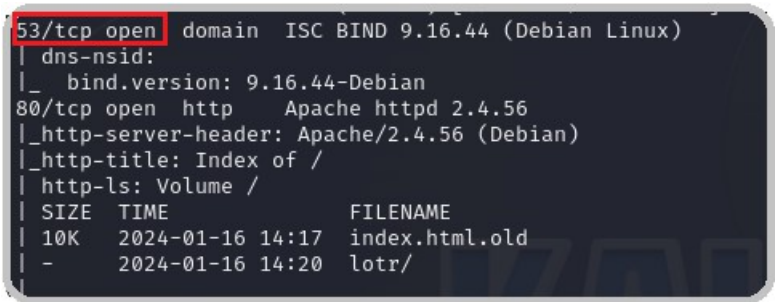
Nous allons tester dans notre cas l'attaque « SYN flood » sur le port http pour empêcher l'accès au serveur web. Le pirate envoie énormément de paquets sur le poste serveur ciblé et son port, ici 80. Le serveur doit répondre à chaque requête ce qui à terme peut le submerger et bloquer momentanément l'accès au serveur web.
Généralement, le pirate se sert de botnet au vue des capacités actuelles des serveurs mais dans notre cas, nous allons attaquer seulement avec la machine Kali Linux.

Exemple d'attaque de déni de service sur le port 80 (HTTP)

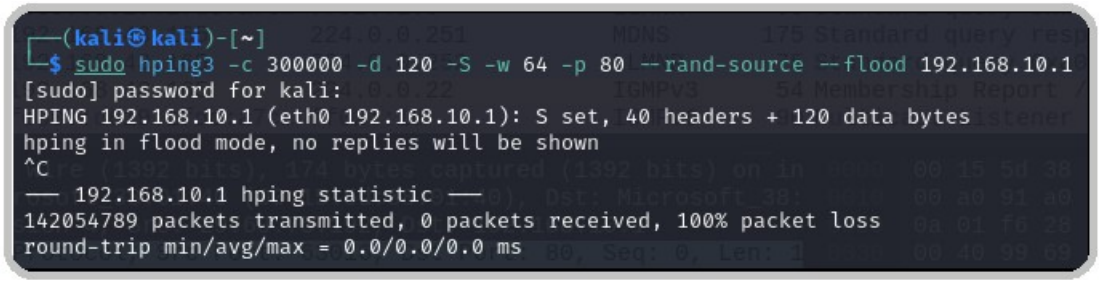
Pour cela, nous allons utiliser l'outil Hping3. C'est une application qui s'exécute par terminal sous Linux. Elle permet l'envoi de paquets TCP, UDP. Elle sert à tester des réseaux et des hôtes mais en premier lieu, lors de sa création, elle était surtout utilisée pour détecter des problèmes de cybersécurité. Hping3 peut envoyer des paquets via un port spécifique, masquer l'adresse IP source, émettre un nombre de paquets très important et bien d'autre grâce à un grand nombre de paramètres.
Elle est déjà installée sur Kali Linux.

Avec cette application, nous allons effectuer une attaque sur le port 80. Cela aura pour conséquence que personne ne pourra accéder à l'interface internet PFsense.

- Avant d'initier l'attaque, nous utilisons la commande nmap pour vérifier que le port 80 est bien ouvert



- Commande de l'attaque



- Wireshark

No.	Time	Source	Destination	Protocol	Length	Info
3708…	2.052360070	211.4.217.246	192.168.10.1	TCP	174	40663 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a reassembled …
3708…	2.052362170	140.125.140.159	192.168.10.1	TCP	174	40664 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a reassembled …
3708…	2.052364255	16.43.49.141	192.168.10.1	TCP	174	40665 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a reassembled …
3708…	2.052366373	70.32.205.149	192.168.10.1	TCP	174	40666 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a reassembled …
3708…	2.052368474	47.70.134.144	192.168.10.1	TCP	174	40667 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a reassembled …
3708…	2.052370613	106.141.68.4	192.168.10.1	TCP	174	40668 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a reassembled …
3708…	2.052373816	200.159.159.32	192.168.10.1	TCP	174	40669 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a reassembled …
3708…	2.052375931	15.21.49.143	192.168.10.1	TCP	174	40670 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a reassembled …
3708…	2.052378019	66.189.205.63	192.168.10.1	TCP	174	40671 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a reassembled …
3708…	2.052380373	150.216.66.15	192.168.10.1	TCP	174	40672 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a reassembled …
3708…	2.052382722	222.71.132.92	192.168.10.1	TCP	174	40673 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a reassembled …
3708…	2.052384907	217.165.45.1	192.168.10.1	TCP	174	40674 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a reassembled …
3708…	2.052387001	160.140.217.22	192.168.10.1	TCP	174	40675 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a reassembled …

- Suricata

Last 250 Alert Entries. (Most recent entries are listed first)

Date	Action	Pri	Proto	Class	Src	SPort	Dst	DPort	GID:SID	Description
06/10/2024 14:21:25	⚠	3	TCP	Not Assigned	250.37.217.242	22288	192.168.10.1	80	1:100036	DOS possible
06/10/2024 14:20:31	⚠	3	TCP	Not Assigned	188.63.189.219	43083	192.168.10.1	80	1:100036	DOS possible
06/10/2024 14:19:40	⚠	3	TCP	Not Assigned	7.215.87.29	58882	192.168.10.1	80	1:100036	DOS possible
06/10/2024 14:18:50	⚠	3	TCP	Not Assigned	31.164.28.67	8125	192.168.10.1	80	1:100036	DOS possible
06/10/2024 14:18:04	⚠	3	TCP	Not Assigned	124.67.121.69	18034	192.168.10.1	80	1:100036	DOS possible
06/10/2024 14:17:17	⚠	3	TCP	Not Assigned	39.198.37.60	48999	192.168.10.1	80	1:100036	DOS possible
06/10/2024 14:16:31	⚠	3	TCP	Not Assigned	46.203.111.92	47892	192.168.10.1	80	1:100036	DOS possible
06/10/2024 14:15:41	⚠	3	TCP	Not Assigned	73.102.23.232	21451	192.168.10.1	80	1:100036	DOS possible